3. Deepa

Program Name: Deepa.java **Input File: deepa.dat**

April is closer than you think! Deepa wants to take care of her taxes ahead of time. The United States, like most countries, has a **progressive tax system**, where the tax rate increases with income. To compute how much she has to pay in taxes, Deepa must look up the year's tax brackets.

A tax bracket is a table with dollar values and tax rates for incomes above the dollar values. This is best explained by example. Consider the following simple tax bracket:

$0.00 10%

$9,525.00 12%

$38,700.00 22%

This means that the first $9,525.00 of income will be taxed at 10 percent. Additional income up to $38,700.00 will be taxed at 12%, and any income beyond that is taxed at 22%. Say Deepa's income is $40,000.00. Then the tax she must pay is:

($9,525.00 * 0.10) + ($29,175.00 * 0.12) + ($1,300.00 * 0.22) = $4,739.50

Since she paid $4,739.50 out of her income of $40,000.00, her "effective" tax is approximately 11.849%. Given the tax table for the year and Deepa's income, how much will she owe in taxes. Furthermore, what is her effective tax rate?

Input: The first line of input has an integer T, the number of test cases. Each test case begins with positive integers N and X, the number of tax brackets and Deepa's taxable income for the year. The next N lines each have two integers, the value in the bracket and the amount by which it is taxed. The brackets are given in ascending order of both dollar value and tax amount. The value in the first bracket is always $0.00. There are at most 20 tax brackets. All dollar values are nonnegative and under $1,000,000.00. All tax percentages are expressed as whole numbers under 100.

Output: For each test case, output the amount she must pay in taxes and the effective tax rate. Include a dollar sign and comma separators for the amount she must pay. Always display two decimal places for cents using normal rounding rules. Display the tax rate rounded to exactly 3 decimal places using standard rounding rules. Format the output as in the sample.

Sample input:

```
2
3 40000
0 10
9525 12
38700 22
4 9823
0 10
9525 12
38700 22
82500 24
```

Sample output:

```
Case #1: $4,739.50 11.849%
Case #2: $988.26 10.061%
```